

## Charter/Spectrum Full Stack Code Challenge

For this challenge we would like you to create a React application and an API that uses restaurant data from the attached JSON file. The React application should present the data in a table and allows users to filter that data.

### User stories are as follows:

- A user should be able to see a table with the name, city, state, phone number, and genres for each restaurant.
- A user should see results sorted by name in alphabetical order starting with the beginning of the alphabet
- A user should be able to filter restaurants by state.
- A user should be able to filter by genre.
- State and Genre filters should default to "All" and take effect instantaneously (no additional clicks).
- A user should be able to enter text into a search field. When hitting the enter key or clicking on a search button, the table should search results. Search results should match either the name, city, or genre.
- A user should be able to clear the search by clearing the text value in the search input.
- A user should only see 10 results at a time and the table should be paginated.
- A user should be able to combine filters and search. The user should be able to turn filters on and off while a search value is present.
- If any of the filters do not return any restaurants, the UI should indicate that no results were found.

### What we are looking for:

- No use of third-party component libraries for the table/filter/search.
- Using Create-React-App or Next.js as a starter kit is okay.
- The API layer should use either Node, Java, or Python. It can be a REST API or GraphQL.
- Filtering and pagination can be done on the API or Front-End layer. Do what you think is best.
- Well organized file structure
- Descriptive naming conventions
- DRY code that is readable and production ready
- Reusable components
- Sound logic for how the filters are architected
- Styling follows a convention/pattern and is well organized
- Full Git history with atomic commits

### Stretch goals:

- Deployed application
- CI / CD
- Unit tests
- TypeScript
- Table row click shows additional information
- User can sort the data by name and state
- Add filter for attire
- Feel free to get creative!